Task 2: software design and development (part A)

A new coffee shop is organising an event for its opening day. At this event, a lucky-dip promotion will be available, where customers can win a discount off their bill.

Below is the analysis and design for a program to calculate customers' bills:

Program analysis
A program is required to calculate a customer's bill. The user will enter the number of items on the bill and then enter the item type for each item (coffee, tea or biscuit). The program will calculate the bill. The bill can then be reduced by using a random value from 1 to 10:

- random value = 1 the customer pays nothing
- random value = 2 to 6 the customer pays half the bill
- random value = 7 to 10 the customer pays the full bill

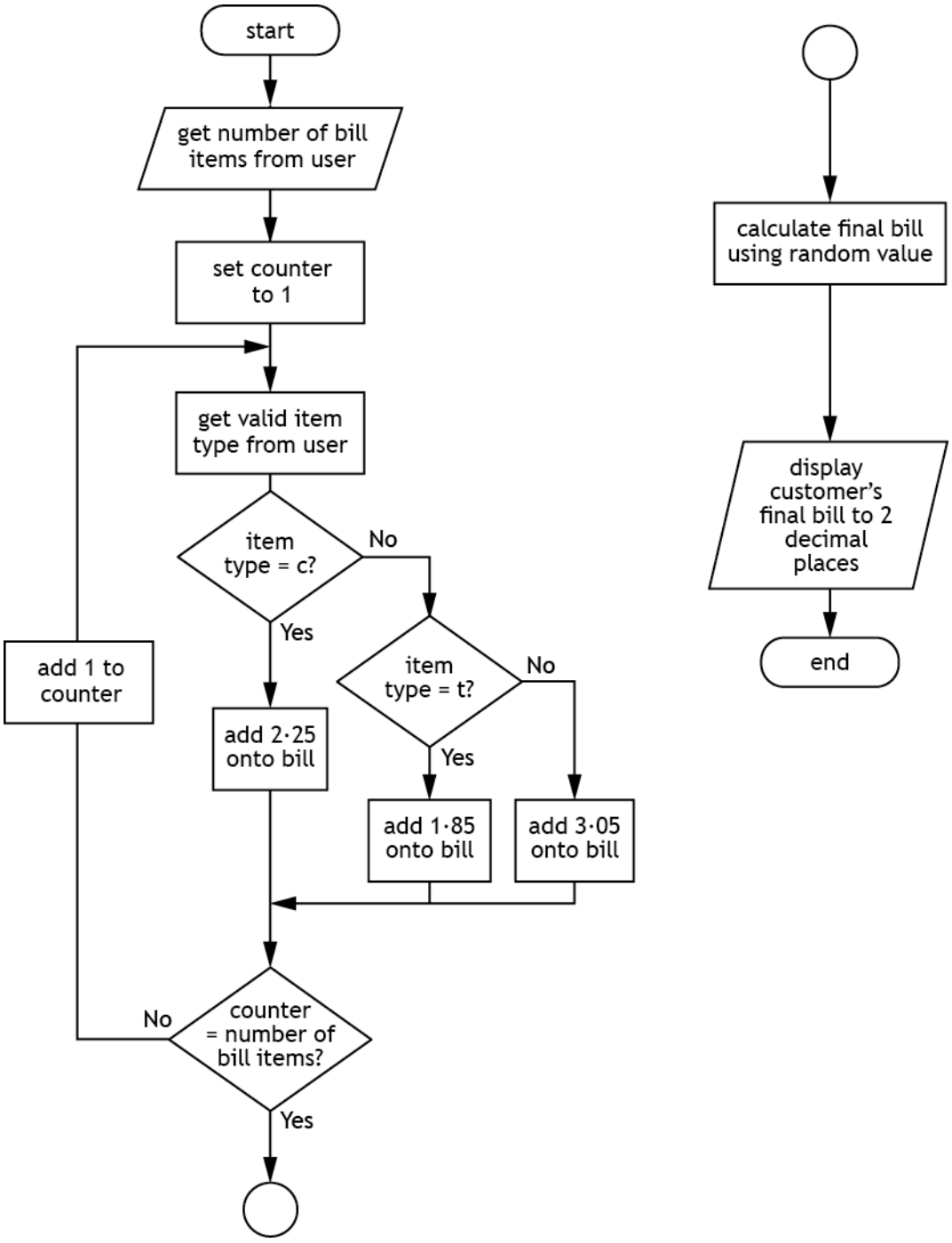
Assumptions
- any number of items can be entered by the user

Inputs
- the number of items on the bill
- the item type for each item on the bill
  - c = coffee
  - t = tea
  - b = biscuit

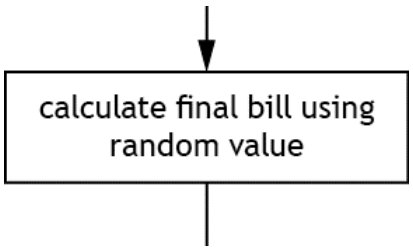
Processes
- generate a random value between 1 and 10
- calculate the total cost of the items on the bill where:
  - coffee = £2.25
  - tea = £1.85
  - biscuit = £3.05
- use the random value to calculate the final bill

Outputs
- the random value
- the cost of the final bill

Program design (flowchart)



2a The flowchart contains the following process:



Using the information provided in the program analysis, expand the design to show how this process could be carried out. You can use a flowchart, structure diagram or pseudocode design.

(3 marks)

1 mark each for design including:

- assignment: random number between 1 and 10
- selection: if random number = 1, set customer's bill to 0
- selection: if random number >= 2 and <=6, set customers bill to 50%

Example Pseudocode

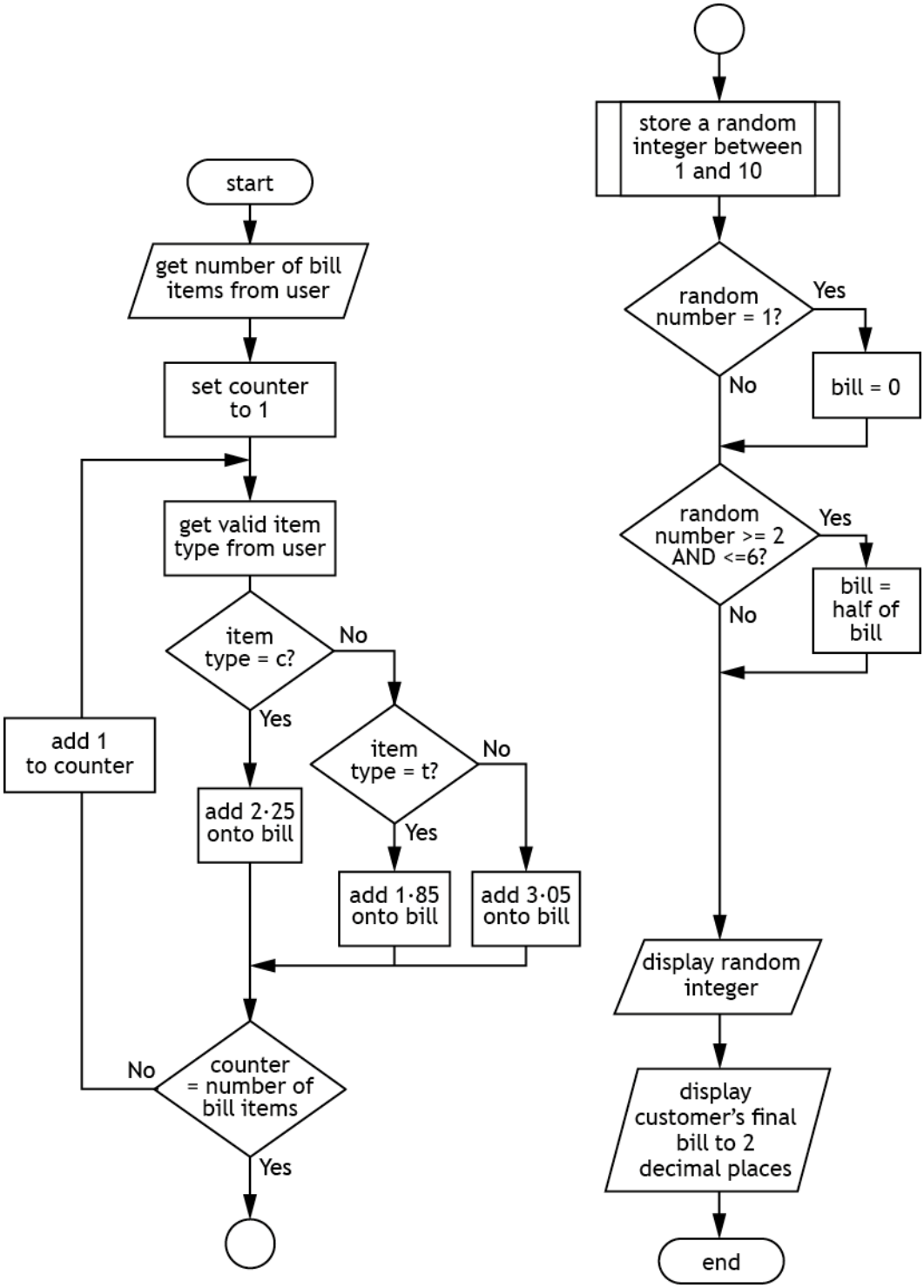
```
1. Generate random number(random) between 1 and 10
2. If random = 1 then
3.     Finalbill = 0
4. End if
5. If random >=2 and <= 6 then
6.     Finalbill = Finalbill * 0.5
7. End if
```

- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_____ Candidate number_____

Task 2: software design and development (part B)

Program design (completed flowchart)



2b Using the program analysis and the design, implement the program in a language of your choice.

Ensure the program matches the flowchart provided on page 17.

(15 marks)

Print evidence of your program code.

```
#2020 Example Solution

noOfItems = 0
bill = 0.0
itemType = ""
randomNo = 0

import random #used to generate random number

noOfItems = int(input("Please enter number of items on bill"))

counter = 0

while counter != noOfItems: #continue to loop for all items
  counter = counter + 1 #add 1 onto counter for each new item.

  #Get valid item type
  itemType = input("Please enter the type of item. c = coffee, t = tea, b = biscuit")
  itemType = itemType.lower()
  while not(itemType == "c" or itemType == "t" or itemType == "b"):
    itemType = input("Invalid - Reenter c = coffee, t = tea, b = biscuit")
    itemType = itemType.lower()

  #add prices for each type of item
  if itemType == "c":
    bill = bill + 2.25
  elif itemType == "t":
    bill = bill + 1.85
  else:
    bill = bill + 3.05


#generate random number between 1 and 10
randomNo = random.randint(1,10)

#Lucky draw - set the bill amount depending on random number
if randomNo == 1:
  bill = 0
  print("Lucky you - you pay nothing")
elif randomNo >=2 and randomNo <=6:
  bill = bill/2
  print("Your bill has been halved")
else:
  print("Unlucky - you pay full price")

#Output final bill and random number
print("-------------------------------------")
print("The random number is " + str(randomNo))
print("The final bill is " + str(round(bill,2)))

```

2c (i) Your program should be tested to ensure it produces one of three different random outputs.

Use the following data to do this:

Number of items: 4
Item 1: coffee
Item 2: tea
Item 3: tea
Item 4: biscuit

```
Please enter number of items on bill 4
Please enter the type of item. c = coffee, t = tea, b = biscuit c
Please enter the type of item. c = coffee, t = tea, b = biscuit t
Please enter the type of item. c = coffee, t = tea, b = biscuit t
Please enter the type of item. c = coffee, t = tea, b = biscuit b
Your bill has been halved
-----------------------------------
The random number is 2
The final bill is 4.5
```

State the possible values (outputs) for the final bill produced from this test data.

Three possible customer bill outputs are:
- £0 bill if number generated a 1
- 4.5 if number generated between 2 and 6 inclusive
- 9 if number over 6

Run your program to show that it produces one of these three outputs.

Print evidence of the test run showing inputs and outputs.

(2 marks)

(ii) Complete the test table below to check the validation for the item type.

Type of test	Test data	
Normal	Number of items – 3	c, t, b
Exceptional	Number of items - 1	y

Run your program to show the result of the exceptional test data.

(1 mark)

Print evidence of the test run.

```
Please enter number of items on bill 1
Please enter the type of item. c = coffee, t = tea, b = biscuit y
Invalid - Reenter c = coffee, t = tea, b = biscuit t
Unlucky - you pay full price
-----------------------------------
The random number is 9
The final bill is 1.85
```

2d With reference to your code, evaluate your program by commenting on the following:

Efficiency of your program code **(2 marks)**
My code is efficient as I used a while loop and within that kept a running total to add new items onto the bill. This was provided as part of the design – the same result could also have been achieved using a fixed loop as the user was asked how many items they would like to add. This may have provided a more efficient solution that a conditional loop in this instance. I also used an IF…ELIF….ELSE to reduce the number of IF statements to 2 instead of 3 which means the processor has less comparisons to make. As there were 3 products, each with different prices this was a more efficient solution. I also used the .lower function to change user input to lowercase. Although not in the N5 course, this saved the need to include a complex condition for both T or t for example.
Robustness of your completed program **(1 mark)**
I have tested my program using Normal, Extreme and Exceptional data and it coped well with unexpected inputs. Normal test data – c, t, b and the program handled these values as expected. Exceptional test data. I tested my program using unexpected data e.g “y” which displayed an error message and asked the user to re-enter. I also ran my program several times to ensure the 3 random scenarios were generated – no bill, half price bill and full bill

Readability of your code **(1 mark)**

My code is readable as I have used good programming techniques.
I have used:

- Meaningful variable names
- Good use of white space. Python uses indentation by default but this also helps readability.

Internal commentary throughout to explain the main parts of my code. This could help with maintenance in the future.

Candidate name_____ Candidate number_____